VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MATEMATIKY
INSTITUTE OF MATHEMATICS

APLIKACE ALGORITMŮ PRO PLÁNOVÁNÍ V ŽELEZNIČNÍ NÁKLADNÍ DOPRAVĚ
APPLICATIONS OF SCHEDULING ALGORITHMS IN RAILWAY TRANSPORTATION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **MICHAL FAJMON**
AUTHOR

VEDOUCÍ PRÁCE **Ing. MARTIN PAVLAS, Ph.D.**
SUPERVISOR

BRNO 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ

Zadání bakalářské práce

Ústav:	Ústav matematiky
Student:	**Michal Fajmon**
Studijní program:	Aplikované vědy v inženýrství
Studijní obor:	Matematické inženýrství
Vedoucí práce:	**Ing. Martin Pavlas, Ph.D.**
Akademický rok:	2017/18

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Aplikace algoritmů pro plánování v železniční nákladní dopravě

Stručná charakteristika problematiky úkolu:

Student si nastuduje problematiku dopravních úloh se zaměřením na kombinatorické problémy typu "vehicle routing problem, job–shop scheduling". Nastudované teoretické poznatky povedou k analýze a návrhu modelu pro plánování dopravy. Ten bude následně implementován do vhodného programovacího prostředí. Vytvořený výpočtový nástroj bude testován na reálných datech z oblasti železniční nákladní dopravy v ČR. Součástí výpočtu bude analýza citlivosti pro klíčové neurčité parametry. Výsledky student zpracuje do požadovaného formátu s ohledem na cíle práce.

Cíle bakalářské práce:

Nastudování problematiky routingových a plánovacích algoritmů
– seznámení se s problematikou železniční nákladní dopravy v ČR
– určení klíčových aspektů, které mají vliv na formulaci modelu pro předmětnou oblast
– diskuse využití modelu pro optimalizaci kombinované dopravy

Seznam doporučené literatury:

GHIANI, Gianpaolo, Gilbert LAPORTE a Roberto MUSMANNO. Introduction to logistics systems management. 2. vyd. ISBN 978-1-119-94338-9.

WILLIAMS, H. Paul. Model building in mathematical programming. 5. vyd. Hoboken, N.J.: Wiley, 2013. ISBN 978-1-118-44333-0.

KARLOF, John K. Integer programming: theory and practice. Boca Raton: Taylor & Francis, 2006. ISBN 978-0-8493-1914-3.

VEELENTURF, Lucas P., Leo G. KROON a Gábor MARÓTI. Passenger oriented railway disruption management by adapting timetables and rolling stock schedules. (Transportation Research Part C: Emerging Technologies), 2017. DOI: 10.1016/j.trc.2017.04.012.

WOLSEY, Laurence A. Integer programming. New York: Wiley,1998. ISBN: 978-0-471-28366-9.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2017/18

V Brně, dne

L. S.

prof. RNDr. Josef Šlapal, CSc. ředitel ústavu	doc. Ing. Jaroslav Katolický, Ph.D. děkan fakulty

Abstrakt
Bakalárska práca je zameraná na vytvorenie matematického modelu, ktorý využíva kombinovanú dopravu pre zvoz odpadu na úrovni obcí. Model je zameraný na železničnú dopravu. Dôvodom tohoto zamerania je nižšia produkcia emisií a zníženie zaťaženia cestnej dopravy. Pre vytvorenie modelu sú použité znalosti z celočíselného a lineárneho programovania a teórie grafov. Výsledný model je implementovaný v prostredí GAMS a testovaný na malej úlohe. Na záver sú uvedené dalšie možnosti vývoja modelu.

Summary
Bachelor's thesis is focused on creation of mathematical model, which uses multiple methods for waste collection at the level of municipalities. Model is focused on rail freight transport. Main reasons are of lower production of emissions and reduction of burden on road traffic. In order to create model, the knowledge of graph theory, integer and linear programming is used. Created model is implemented in GAMS environment on small scale data. At the end there are stated possibilities for further development of the model.

Kľúčové slová
celočíselné programovanie, tok v sieti, GAMS, zvoz odpadu, železničná nákladná doprava

Keywords
integer programming, network flow, GAMS, waste transport, rail freight transport

FAJMON, M. *Aplikace algoritmů pro plánování v železniční nákladní dopravě.* Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2018. 23 s. Vedoucí Ing. Martin Pavlas, Ph.D.

Prehlasujem, že som bakalársku prácu na tému „Aplikace algoritmů pro plánování v železniční nákladní dopravě“ vypracoval samostatne s použitím literatúry uvedenej v zozname.

Michal Fajmon

Ďakujem vedúcemu práce Ing. Martinovi Pavlasovi Ph.D za korektúru a pripomienky k práci. Ďakujem Ing. Vlastimírovi Nevrlému za ochotu, rady a čas pri konzultáciách. Ďalej by som rád poďakoval Ing. Radovanovi Šomplákovi, Ph.D. za rady pri vytváraní modelu a Ing. Jiřímu Gregorovi za konzultácie ohľadom železničnej dopravy.

Michal Fajmon

Obsah

Úvod

Odpad je neoddeliteľnou súčasťou nášho každodenného života, preto sa s rastom populácie zvyšuje aj dôležitosť fungovania udržateľného odpadového hospodárstva. Odpad je možné spracovať alebo odstrániť, v každom prípade je však nutné ho premiestniť.

Táto práca sa zaoberá modelovaním transportou odpadu so zameraním na železničnú dopravu. Elektrická frakcia železničnej nákladnej dopravy je z pohľadu emisií podstatne vhodnejšia ako cestná nákladná doprava [3]. Vhodným zapojením železničnej dopravy do zvozu odpadu by sa mohlo taktiež znížiť zaťaženie cestnej dopravy. S prepravou pomocou železničnej dopravy sú však spojené rôzne komplikácie.

Cieľom práce bolo vytvoriť matematický model, ktorý by popisoval zvoz odpadu niekoľkými prostriedkami vrátene železničnej nákladnej dopravy. Prvá časť práce je venovaná matematickému programovaniu, všeobecnej formulácii problémov lineárneho celočíselného programovania a riešeniu vzorovej úlohy. V nasledujúcej kapitole sú uvedené konkrétne príklady použitia celočíselného programovania. Pri jednotlivých problémoch sú uvedené použiteľné algoritmy a ich výpočtová náročnosť.

Pri vytvorení modelu bola uvažovaná kombinácia cestnej a železničnej dopravy s možnosťou výstavby prekladacích zariadení, ktoré umožnia transport odpadu po železnici. Na záver je vytvorený model testovaný na malej úlohe a jej riešenie je okomentované. V práci bude využívaná anglická terminoloógia a skratky z dôvodu absencie niektorých termínov v slovenskom jazyku a následného zachovania konzistentnosti názvov.

1 Matematické programovanie

Matematické programovanie je oblasť matematiky zaoberajúca sa teóriou a metódami pre riešenie problémov nájdenia extrémov funkcií na množine. Táto množina je definovaná lineárnymi a nelineárnymi obmedzeniami v konečne rozmernom vektorovom priestore. Aplikácie môžme nájsť v rôznych oblastiach, ako je napríklad plánovanie výroby, doprava, poľnohospodárstvo, hutníctvo, atď.. Hľadaný typ extrému je rôzny pre konkrétne problémy. V niektorých hľadáme riešenie zabezpečujúce najmenšie straty, v iných riešenie maximalizujúce zisk. Formuláciu problému uvedieme ako minimalizačnú. Hľadáme minimum funkcie $z(\mathbf{x})$ kde $\mathbf{x}$ je n-rozmerný vektor patriaci do množiny X.

$$X = \{x : q_i(\mathbf{x}) \geq 0, i = 1, ..., k;\ h_i(\mathbf{x}) = 0, j = 1, ..., m\}, \tag{1.1}$$

kde $q_i(\mathbf{x})$ a $h_i(\mathbf{x})$ sú funkcie zobrazujúce $\mathbb{R}^n \to \mathbb{R}$. Funkciu $z(\mathbf{x})$ nazývame účelová funkcia a X množinou prípustných riešení. V matematickom programovaní môžme rozlíšiť oblasti ako lineárne programovanie, celočíselné programovanie, zmiešané celočíselné programovanie, kombinatorické programovanie, ktoré budú v nasledujúcich častiach podrobnejšie popísané.

V mnohých prípadoch sa pri modelovaní využívajú pojmy z teórie grafov. Najzákladnejším pojmom je orientovaný graf. Je to trojica $G = (V, E, \epsilon)$, kde V označuje konečnú monžinu vrcholov, E je množina orientovaných hrán a ϵ je zobrazením $\epsilon : E \to V^2$ nazývaným vzťah incidencie. Toto zobrazenie priradí každej hrane usporiadanú dvojicu vrcholov (x, y) $x, y \in V$, kde x je počiatočný vrchol a y je koncový vrchol. Takýto graf je však často nepostačujúci k popisu situácie a preto sa priraďujú hranám a vrcholom hodnoty, ktoré môžu predstavovať vzdialenosť, obtiažnosť, atď.. Na základe týchto poznatkov je možné zaviesť incidenčnú maticu, ktorá priradí dvojici vrchol, hrana hodnotu 1 ak je vrchol koncovým vrcholom hrany, -1 ak je začiatočným vrcholom hrany a 0 v ostatných prípadoch. V prípade modelu zvozu odpadu v tejto práci predstavujú hodoty priradené hranám vzdialenosť medzi vrcholmi(uzlami).

1.1 Lineárne programovanie

V lineárnych modeloch musia byť všetky uvažované rovnice a nerovnice lineárne vzhľadom k rozhodovacím premenným. Každý lineárny model je možné prepísať na štandardný tvar. Jednotný tvar je potrebný pre uplatnenie algoritmu na riešenie modelu, akým je simplexová metóda. Vlastnosti štandardného tvaru sú nasledovné:

- Úlohou je maximalizovať účelovú funkciu.
- Všetky premenné sú nezáporné.
- Všetky obmedzenia, až na tie zabezpečujúce nezápornosť premenných, sú v tvare rovníc s nezápornou pravou stranou.

1.2 Spôsob zápisu

Úlohu lineárneho programovania je možné zapísať nasledovne. Nech $\mathbf{x} = (x_j)$ a $\mathbf{c} = (c_j)$ sú vektory a $\mathbf{A} = (a_{ij})$ je matica $m \times n$ pričom platí $i = 1, 2, ..., m$; $j = 1, 2, ..., n$ a $a_{i,j}, c_j, x_j \in \mathbb{R}$. Pri uvedenom značení je zápis nasledovný.

$$max\ \mathbf{c}^T\mathbf{x} \tag{1.2}$$

$$\begin{aligned} \text{s.t.} \quad \mathbf{A}\mathbf{x} &\leq \mathbf{b} && (1.3) \\ \mathbf{x} &\geq 0 && (1.4) \end{aligned}$$

Množina všetkých bodov spĺňajúcich sústavu nerovníc sa nazýva množina prípustných riešení, pre lineárny model je vždy konvexná. Hovoríme, že množina $\mathbf{S}$ je konvexná ak pre každé 2 body $X, Y \in S$ a $t \in \{0, 1\}$ platí, že $(X - Y)t + Y \in S$. Ďalším spôsobom zápisu je tzv. sumačne indexový zápis.

$$max \sum_{j=1}^{n} c_j x_j \tag{1.5}$$

$$\begin{aligned} \text{s.t.} \quad \sum_{j=1}^{n} a_{i,j} x_j &\leq b_i, && \forall i \in \{1, ..., m\} && (1.6) \\ x_j &\geq 0, && \forall j \in \{1, ..., n\} && (1.7) \end{aligned}$$

Tento typ zápisu je vhodný pre softvér akým je napríklad GAMS (General Algebraic Modeling System), AMPL (A Mathematical Programming Language). Pre MATLAB je vhodnejší prvý uvedený zápis pomocou matíc a vektorov.

1.3 Celočíselné programovanie

Pre mnoho praktických problémov je nutné aby niektoré premenné nadobúdali iba celočíselné hodnoty. Modely obsahujúce výhradne celočíselné premenné patria do skupiny PIP (pure integer programming). Avšak bežnejšou skupinou sú modely obsahujúce celočíselné a zároveň spojité premenné. Tieto modely patria do MIP (mixed integer programming). Očakávané použitie celočíselnosti, zabezpečiť celočíselnosť premenných predstavujúcich nedeliteľné objekty, nie je úplne bežné. Vo väčšine praktických problémov sú celočíselné premenné obmedzené na 0 a 1. Tieto modely sú súčasťou BP (binary programming). Nasledujúci model je lineárny model zmiešaného celočísleného programovania a má tvar

$$max\ \mathbf{c}^T\mathbf{x} + \mathbf{d}^T\mathbf{y} \tag{1.8}$$

$$\begin{aligned} \text{s.t.} \quad \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} &\leq \mathbf{b} && (1.9) \\ \mathbf{x} &\geq \mathbf{0} && (1.10) \\ \mathbf{y} &\in \mathbb{Z}_+^p && (1.11) \end{aligned}$$

kde $\mathbf{x} = (x_j)$, $\mathbf{y} = (y_j)$, $\mathbf{b} = (b_i)$ a $\mathbf{c} = (c_j)$ sú vektory, $\mathbf{A} = (a_{i,j})$, $\mathbf{B} = (b_{i,j})$ sú matice $m \times n$, pričom platí $i = 1, 2, ..., m$; $j = 1, 2, ..., n$ a $a_{i,j}, b_{i,j}, y_j, c_j, x_j \in \mathbb{R}$. Predchádzajúcu formuláciu je možné jednoducho zmeniť na model PIP vynechaním $\mathbf{c}^T\mathbf{x}$ a $\mathbf{Ax}$, prípadne na model BP zmenou množiny $\mathbb{Z}_+^p$ na $\{0,1\}^p$. Problém s celočíselnými modelmi je ich výpočetná náročnosť. Pre riešenie IP modelu je potrebných mnohonásobne viac operácii ako pre model so spojitými premennými porovnateľnej veľkosti. Pre riešenie celočíselných modelov sa používajú algoritmy, ktoré môžme rozdeliť na dve skupiny, exaktné a heuristické. Medzi exaktné patria algoritmy ako *Branch and Bound, Cutting plane* alebo *Branch and cut*, ktorý je kombináciou dvoch predchádzajúcich. Heuristické algoritmy sa používajú najmä pre svoju rýchlosť. Aj keď výsledkom nemusí byť globány extrém, často riešenie vyhovuje ak poskytuje dostatočné zlepšenie aktuálneho známeho prípustného riešenia. Medzi tieto algoritmy patria napríklad *Tabu search, Hill climbing, Simulated annealing.*

1.3.1 Kombinatorické problémy

Termínom kombinatorický problém označujeme úlohu, ktorej charakteristikou je značne veľký počet prípustných riešení. Ich počet vychádza napríklad z rôzneho poradia vykonávaných operácii alebo priraďovania ľudí na rôzne pozície. Podľa uvedených príkladov môžme ďalej tieto úlohy rozdeliť na *allocating problems* a *sequencing problems.* Riešenie týchto problémov nie je môžné získať v relevantnom čase pomocou úplnej enumerácie z dôvodu veľkého množstva možných riešení. Kombinatorický problém je možné zapísať v tvare

$$\min_{S \subseteq N} \{ \sum_{j \in S} c_j : S \in F \}$$

kde N je konečná množina $N = \{1, 2, ..., n\}$, c_j $\forall j \in N$ sú váhy a F je množina prípustných podmnožín N. Algoritmy používané na riešenie týchto problémov môžme rozdeliť na deterministické ako napríklad *branch and bound* a nedeterministické heuristiky ako *Tabu search, Simulated annealing*, a rôzne genetické algoritmy. Pre deterministické algoritmy platí, že pri rovnakých vstupných údajoch dospejú za rovnaký čas vždy k rovnakému výsledku. Všetky zmienené algoritmy v tejto časti sú popísane v [9].

1.4 Simplexová metóda

Simplexová metóda je jedným z algoritmov pre riešenie úloh lineárneho programovania. Opodstatnenie tejto metódy nájdeme aj pri celočíselnom programovaní konkrétne pri výpočte relaxovaných MIP modelov. Relaxácia znamená uvoľnenie niektorých požiadaviek, pri celočíselných modeloch ide o odstránenie požiadavky na celočíselnosť premenných. V prípade maximalizačných úloh je maximálna hodnota účelovej funkcie relaxovaného problému väčšia nanajvýš rovná hodnote účelovej funkcie nerelaxovaného problému. Pri LP(linear programming) modeloch tvorí množinu prípustných riešení konvexný mnohosten, ktorý môže byť neohraničený. Simplexový algoritmus spočíva v presune po vrcholoch mnohostenu pričom sa zlepšuje hodnota účelovej funkcie.

Simplexová metóda je dvojkroková. Cieľom prvého kroku je nájsť štartovací vrchol. V závislosti na type problému riešenie môže byť triviálne, ale vo všeobecnosti môže byť riešenie nájdené pomocou simplexového algoritmu uplatneného na upravenú úlohu.

Výsledok prvého kroku je buď prípustné riešenie, alebo je množina prípustných riešení prázdna. V druhom kroku je použitý algroritmus so začiatočným bodom získaným v prvom kroku. Výsledok druhého kroku je optimálne riešenie alebo nekonečná hrana, na ktorej je hodnota účelovej funkcie. Aby bolo možné riešiť úlohy pomocou simplexovej metódy je potrebné previesť úlohu na štandardný tvar, ktorý je pomocou sumačne indexového zápisu zapísaný nasledovne.

$$max \sum_{j=1}^{n} c_j x_j \tag{1.12}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} a_{i,j} x_j = b_i, \qquad \forall i \in \{1, ..., m\} \tag{1.13}$$

$$x_j \geq 0, \qquad \forall j \in \{1, ..., n\} \tag{1.14}$$

V tomto prípade je n počet premenných a m počet rovníc.

1.5 Výpočtová zložitosť

Úlohy matemetického programovania možno rozčleniť na skupiny podľa náročnosti ich výpočtu. Najjednoduchšou skupinou na riešenie sú problémy patriace do $\mathcal{P}$. Je to skupina rozhodovacích problémov, pre ktoré sme schopní v polynomiálnom čase získať odpoveď ÁNO alebo NIE. Pravdepodobne skupinou, ktorá je náročnejšia na výpočet je $\mathcal{NP}$ (nondeterministic polynomial). Nieje dokázané, že $\mathcal{P} \neq \mathcal{NP}$, platí však $\mathcal{P} \subseteq \mathcal{NP}$. Jednou z možných definícií tejto skupiny je nasledujúca. Dokážeme overiť správnosť dodaného riešenia v polynomiálnom čase, nie však získať riešenie v polynomiálnom čase. Medzi najťažšie doteraz zmienené problémy patria $\mathcal{NP} - complete$. Ak $c \in \mathcal{NP} - complete$ tak platí, $c \in \mathcal{NP}$ a zároveň, každý problém patriaci do $\mathcal{NP}$ je redukovateľný na c v polynomiálnom čase. Skupinu $\mathcal{NP} - hard$ tvoria problémy, ktoré sú aspoň také zložité ako $\mathcal{NP} - complete$. V širšom rozsahu o výpočetnej zložitosti pojednáva [9].

1.6 Ukážkový príklad

Nech existuje továreň na výrobu hračiek. Vyrábajú sa dva druhy Hračiek H1 a H2. Každá z nich musí prejsť cez dva procesy vo výrobe P1 a P2. Budeme uvažovať nevyčerpateľné materiálové zdroje a dopyt taký, že všetko čo sa vyrobí, sa predá. Pri výrobe sme obmedzený časom, ktorý môžme využiť na jednotlivé procesy. Pre P1 je to 82,5 h a pre P2 100 h. Časové náklady a zisk z jednotivých hračiek sú uvedené v tabuľke 1.1.

Tabuľka 1.1: Náklady na výrobu a zisk

	P1	P2	Zisk [Kč]
H1	1	2	3
H2	1	1	2

Našou úlohou je zistiť x_1 počet vyrábaných kusov hračiek H1 a x_2 počet vyrábaných kusov hračiek H2 tak, aby bol zisk maximálny.
Označíme celkový zisk ako z. Je zrejmé že celkový zisk sa vypočíta nasledovne:

$$z = 3x_1 + 2x_2 \tag{1.15}$$

Ďalej musíme pridať obmedzenia týkajúce sa výrobných procesov:

$$x_1 + x_2 \leq 82,5 \tag{1.16}$$
$$2x_1 + x_2 \leq 100 \tag{1.17}$$

Nakoniec pridáme podmienku pre nezápornosť počtu vyrábaných hračiek.

$$x_1, x_2 \geq 0 \tag{1.18}$$

Celú úlohu môžme spolu zapísať nasledovne.

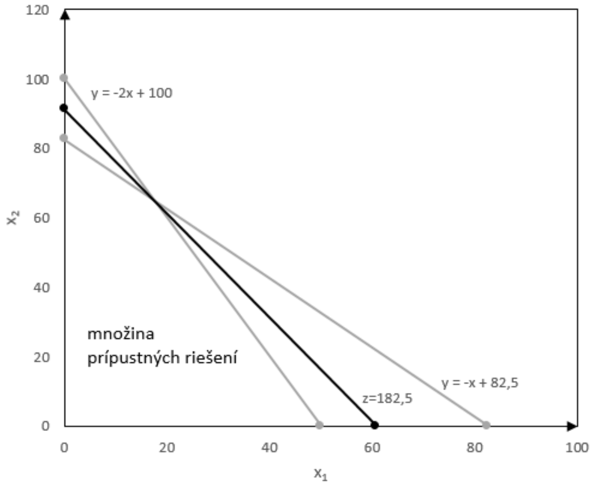
$$max\ z \tag{1.19}$$
$$z = \ 3x_1 + 2x_2 \tag{1.20}$$

$$\text{s.t.} \quad x_1 + x_2 \leq 82,5 \tag{1.21}$$
$$2x_1 + x_2 \leq 100 \tag{1.22}$$
$$x_1, x_2 \geq 0 \tag{1.23}$$

Účelová funkcia nadobúda svoje lokálne maximum v bode $x_1 = 17,5$, $x_2 = 65$. Toto maximum je zároveň globálnym, pretože maximalizujeme konvexnú funkciu na konvexnej monožine. Riešenie môžme interpretovať tak, že výsledný pomer vyrábaných hračiek je $\frac{H1}{H2} = \frac{17,5}{65} = \frac{7}{26}$. Ak však chceme získať maximálny zisk za podmienky, že všetky hračky musia byť dokončené, lebo iba tie sa dajú predať, je nutné obmedziť premenné x_1 a x_2 na celočíselné hodnoty. Optimálne celočíselné riešenie tejto úlohy je $x_1 = 18$, $x_2 = 64$.

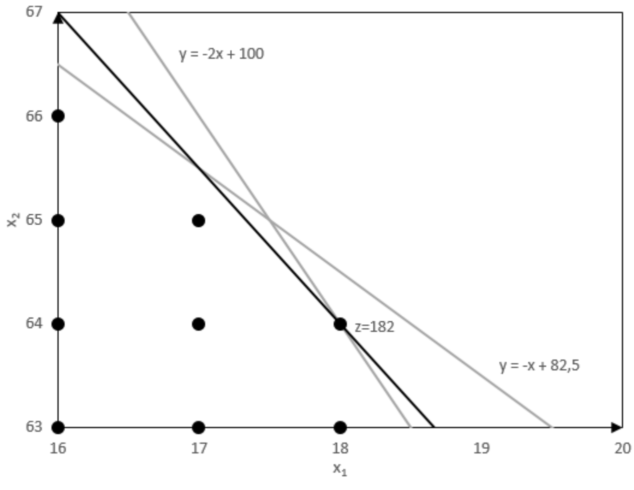
1.6.1 Grafické riešenie

Tento spôsob riešenia nemá pre väčšie úlohy zmysel, pretože ním dokážeme prakticky riešiť len úlohy s dvomi rozhodovacími premennými. Z tohto dôvodu má grafické riešenie skôr ilustračný charakter. Graficky vyriešime úlohu uvedenú v 1.4.
Ako prvé zostrojíme súradnicové osi a riešenie budeme hľadať ako bod (x_1, x_2). Ďalej na základe obmedzení modelu zostrojíme množinu prípustných riešení.



Obr. 1.1: Riešenie úlohy so spojitými premennými

Určíme gradient účelovej funkcie $\nabla z = (\frac{\partial z}{\partial x_1}, \frac{\partial z}{\partial x_2})$. V našom prípade je to vektor $\nabla z = (3, 2)$. Gradient je kolmý na vrstevnice, pozdĺž ktorých je hodnota účelovej funkcie koštantná. Optimálne riešenie má hodnotu vrstevnice s najvyššou hodnotou, ktorá má nenuluvý prienik s množinou prípustných riešení. V našom prípade $z = 182,5$ pre $x_1 = 17,5$ a $x_2 = 65$.
V prípade podmienky celočíselnosti pri premenných x_1, x_2 sa množina prípustných riešení zredukuje na množinu diskrétnych bodov. Tieto body môžme vidieť na Obr. 1.2. Postup pri grafickom riešení je pre celočíselné premenné identický.



Obr. 1.2: Detail oblasti v okolí optima pre úlohu s celočíselnými premennými

1.6.2 Riešenie pomocou simplexovej metódy

Simplexová metóda spočíva v postupnom zlepšovaní prípustného riešenia. Ako prvé je potrebné previesť úlohu na štandardný tvar. Prevod je docielený použitím doplnkových premenných. Štandardný tvar pre úlohu z kapitoly 1.4. vyzerá nasledovne.

$$max\ z \tag{1.24}$$
$$z = \ 3x_1 + 2x_2 \tag{1.25}$$

$$\text{s.t.}\quad x_1 + x_2 + x_3 \qquad \leq 82,5 \tag{1.26}$$
$$2x_1 + x_2 \qquad + x_4 \leq 100 \tag{1.27}$$
$$x_1, x_2, x_3, x_4 \geq 0 \tag{1.28}$$

Doplnkovými premennými sú $x_3,\ x_4$. Prvý krok spočíva v nájdení prípustného riešenia vo vrchole množiny prípustných riešení. V tomto prípade je riešenie triviálne, $x_1 = 0,\ x_2 = 0$.

V druhom kroku postupujeme iteračne, zostrojíme prvú tabuľku simplexového algoritmu viď. tabuľka 1.2 a nájdeme pivota. Ako prvé určíme stĺpec so záporným číslom v prvom riadku s najväčšou absolútnou hodnotou. Ďalej je potrebné určiť riadok, pre ktorý platí, že podiel RHS (right hand side) a čísla vo vybranom stĺpci je minimálny. Prvok, ktorý sa nachádza vo vybranom stĺpci a riadku je pivot.

Tabuľka 1.2: Prvá iterácia simplexového algoritmu s vyznačeným pivotom.

	z	x_1	x_2	x_3	x_4	RHS
z	1	-3	-2	0	0	0
x_3	0	1	1	1	0	82,5
x_4	0	2	1	0	1	100

Tabuľku upravíme pomocou operácií Gausovej eliminačnej metódy tak by na mieste pivota bola hodnota rovná 1 a ostatné hodnoty v stĺpci s pivotom boli rovné 0. Tento postup opakujeme až do chvíle keď prvý riadok obsahuje iba nezáporné hodnoty. Tabuľka 1.3 zobrazuje druhú iteráciu algoritmu.

Tabuľka 1.3: Druhá iterácia simplexového algoritmu s vyznačeným pivotom

	z	x_1	x_2	x_3	x_4	RHS
z	1	0	-0,5	0	1,5	150
x_3	0	0	0,5	1	-0,5	32,5
x_1	0	1	0,5	0	0,5	50

Z výslednej tabuľky 1.4 môžme vyčítať z prvého riadku a posledného stĺpca maximálnu hodnotu účelovej funkcie 182,5. V zvyšných dvoch riadkoch sa nachádzajú optimálne hodnoty rozhodovacích premenných, pri ktorých nadobúda účelová funkcia maximálnu hodnotu.

Tabuľka 1.4: Výsledá tabuľka simplexového algoritmu.

	z	x_1	x_2	x_3	x_4	RHS
z	1	0	0	1	1	182,5
x_2	0	0	1	2	-1	65
x_1	0	1	0	-1	1	17,5

2 Vybrané modely celočíselného programovania

V tejto kapitole sú uvedené niektoré z úloh celočíselného programovania. Rozšírenia a zovšobecnenia týchto úloh môžu byť použité pre popis rôznych dopravných a priraďovacích úloh. Napríklad *vehicle routing problem* je zovšeobecnením TSP (traveling salesman problem). Pri vytváraní vlastného modelu som čerpal z formulácií týchto problémov. Každý z týchto problémov môžme zaradiť medzi kombinatorické problémy.

2.1 Traveling salesman problem

Tomuto problému je venovaná veľká pozornosť z dôvodu jeho koncepčnej jednoduchosti. Na druhej strane riešenie tohto problému je veľmi náročné. Názov pochádza z problému obchodníka, ktorý má navštíviť množinu miest a vrátiť sa domov. Úlohou je nájsť poradie miest tak, aby sa minimalizovala celková prejdená vzdialenosť. Ďalšia možná interpretácia tohto problému môže byť JSP (Job-shop problem), ktorý bude uvedený neskôr v tejto kapitole. Existuje viacero možných formulácií tohto problému, jednou z nich je MTZ (Miller-Tucker-Zemlin), ktorá je uvedená v tejto časti.

$$min \sum_{i=1}^{n} \sum_{i \neq j, j=1}^{n} c_{i,j} x_{i,j} \tag{2.1}$$

$$\begin{aligned}
\text{s.t.} \quad & \sum_{i \neq j, i=1}^{n} x_{i,j}, = 1 && \forall j \in \{1, ..., n\} && (2.2)\\
& \sum_{i \neq j, j=1}^{n} x_{i,j}, = 1 && \forall i \in \{1, ..., n\} && (2.3)\\
& u_i - u_j + 1 \leq (n-1)(1 - x_{i,j}), && \forall i, j : \ 2 \leq i \neq j \leq n; && (2.4)\\
& x_{i,j} \in \{0, 1\}, && \forall i, j \in \{1, ..., n\} && (2.5)\\
& u_i, u_j \in \mathbb{N}, && \forall i, j \in \{1, ..., n\} && (2.6)
\end{aligned}$$

Parameter $c_{i,j}$ predstavuje vzdialenosť medzi mestom i a j. Účelová funkcia (2.1) minimalizuje celkovú prejdenú vzdialenosť Podmienka (2.2) zabezpečuje, že do každého mesta pricestujeme práve raz. Podmienka (2.3) zabezpečuje, že každé mesto opustíme práve raz. Podmienka (2.4) zabraňuje vytváraniu podtrás. Podtrasa je v našom prípade postupnosť miest, ktorá začína a končí v rovnakom mieste ale neobsahuje všetky mestá. Existujú viaceré exaktné algoritmy pre výpočet tohto problému. Výpočtová náročnosť úplnej enumerácie je pre tento problém $O((n-1)!)$. Ak by sme daný problém riešili pomocou *Held-Karp* algoritmu výpočtová je $O(n^2 2^n)$, čo je menej ako pri riešení úplnou enumeráciu pre $n > 8$.

2.2 Set covering problem

Tento problém nadobudol svoje pomenovanie z nasledujúceho abstraktného problému. Nech existuje množina objektov, ktoré očíslujeme ako $S = 1, 2, 3, ..., m$. Ďalej nech existuje trieda $\mathbb{S}$ podmnožín S. Každý člen triedy $\mathbb{S}$ má priradený parameter, cenu. Úlohou je "pokryť" všetky prvky S za čo najnižšiu cenu.

$$min \sum_{i=1}^{n} \delta_i c_i \tag{2.7}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} x_{j,i}\delta_i \geq 1, \qquad \forall j \in \{1, ..., m\} \tag{2.8}$$

$$\delta_i \in \{0, 1\}, \qquad \forall i \in \{1, ..., n\} \tag{2.9}$$

$$\delta_i = \begin{cases} 1 & \text{ak i-ty člen } \mathbb{S} \text{ patrí do pokrytia} \\ 0 & \text{ak i-ty člen } \mathbb{S} \text{ nepatrí do pokrytia} \end{cases}$$

Rovnica (2.8) zabezpečuje aby každý prvok S bol pokrytý aspoň raz. V tomto prípade predstavuje c_i cenu i-teho prvku $\mathbb{S}$ a x_{ji} určujú, ktoré prvky S pokryje i-ty prvok $\mathbb{S}$. Jedna z možných aplikácií je vytvorenie harmonogramu pre posádky letov. Tento problém možno pretvoriť na tzv. vážený. Zmena spočíva v tom, že neobmedzíme pravú stranu podmienky striktne na 1, ale povolíme hodnoty patriace do $\mathbb{Z}^+$. Prvky S, ktoré majú v podmienke na pravej strane nerovnice číslo k>1 musia byť "pokryté" k-krát. Jeden z možných použiteľných algoritmov je *Kuhn–Munkres algorithm*, ktorého výpočtová náročnosť je $O(n^3)$.

2.3 The assignment problem

V tomto prípade je úlohou priradiť n pracovníkov k n úlohám pričom minimalizujeme celkový strávený čas potrebný na dokončenie úloh. Pracovník i schopný vykonať úlohu j strávi pri tomto procse čas t_{ij}. Ak pracovník i nie je schopný vykonať úlohu j, tak za t_{ij} je dosadená dostatočne veľká hodnota. Tento problém možno interpretovať aj ako dopravný problém s n zdrojmi o produkcii 1 a n miestami s dopytom 1.

$$min \sum_{i}^{n} \sum_{j}^{n} t_{i,j} x_{i,j} \tag{2.10}$$

$$\text{s.t.} \quad \sum_{j}^{n} x_{i,j} = 1, \qquad \forall i \in \{1, ..., n\} \tag{2.11}$$

$$\sum_{i}^{n} x_{i,j} = 1, \qquad \forall j \in \{1, ..., n\} \tag{2.12}$$

kde

$$x_{i,j} = \begin{cases} 1 & \text{ak je pracovník i je priradeý k úlohe j} \\ 0 & \text{inak} \end{cases}$$

Účelová funkcia (2.10) minimalizuje súčet jdnotlivých časov strávených pri vykonávaní úloh. Rovnica (2.11) zabezpečuje, že každý pracovník je priradený práve k jednej úlohe. Rovnica (2.12) zabazpečuje, že každá úloha je vykonávaná práve jedným pracovníkom. Aj keď je riešenie tohto problému nutne celočíselné, nie je ho potrebné formulovať ako IP. Riešenie, ktoré získame pomocou metód pre LP je celočíselné, z dôvodu celočíslenosti hodôt pravých strán rovníc (2.11) a (2.12). Simplexová metóda je teda vhodným algoritmom pre výpočet tohto problému.

2.4 Job-shop scheduling

Tento problém je popísaný pomocou n úloh $\{J_o\}_{1\leq o\leq n}$, ktoré môžu byť vykonané na množine m strojov $\{M_k\}_{1\leq k\leq m}$. Každá úloha má technologickú postupnosť strojov $(\sigma_1^k, ..., \sigma_m^k)$ v akom musia byť použité. Symbolom σ_u^k je označená u-ta operácia úlohy k, kde $1 \leq u \leq m$. Každej dvojici úloha i a stroj k je priradená nezáporná celočíselná hodnota $p_{i,k}$, ktorá reprezentuje čas potrebý pre spracovanie na stroji k pre úlohu i. Celočíselná premenná $x_{i,j}$ predstavuje čas začatia úlohy j na stroji i. Ak úloha j predchádza úlohe k na stroji i tak premenná $z_{i,j,k}$ je rovná 1. Účelovou funkciu je v tomto prípade čas potrebný na dokončenie všetkých úloh. Ďalšie požiadavky pre model:

- Na jednom stroji nemôže súčasne prebiehať viacero operácií.
- Každá operácia musí byť vykonaná práve raz.
- Stroje sú vždy dostupné.
- Čas pri prechode medzi operáciami je nulový.

Formulácia popísaného modelu je nasledovná.

$$min\ C_{max} \tag{2.13}$$

$$\begin{aligned}
\text{s.t.}\quad x_{k,i} &\geq 0, && \forall j \in J; \forall i \in M && (2.14)\\
x_{\sigma_h^j,j} &\geq x_{\sigma_{h-1}^j,j} + p_{\sigma_{h-1}^j,j}, && \forall j \in J; \forall h \in \{2, ..., m\} && (2.15)\\
x_{i,j} &\geq x_{i,k} + p_{i,k} - V z_{i,j,k}, && \forall i \in M; \forall j,k \in J : j < k && (2.16)\\
x_{i,k} &\geq x_{i,j} + p_{i,j} - V(1 - z_{i,j,k}), && \forall i \in M; \forall j,k \in J : j < k && (2.17)\\
C_{max} &\geq x_{\sigma_m^j,j} + p_{\sigma_m^j,j}, && \forall j \in J && (2.18)\\
z_{i,j,k} &\in \{0,1\}, && \forall i \in M; \forall j,k \in J && (2.19)
\end{aligned}$$

Rovnica (2.14) zabezpečuje, že začiatočný čas každej úlohy je väčší ako 0. Rovnica (2.15) zabezpečuje, že úlohy sú vykonané v danom poradí. Cieľom rovníc (2.16) a (2.17) je zamedziť, vykonávaniu viac ako jednej operácie na jednom stroji v tom istom čase. Konštanta V musí mať dostatočne veľkú hodnotu aby zabezpečovala korektnosť rovníc. Rovnica (2.18) zabezpečuje, že C_{max} (čas potrebný na dokončenie všetkých úloh) je aspoň tak veľký ako čas dokončenia poslednej operácie každej úlohy. V roku 1979 bolo dokázané, že pre $m > 3$ tento problém patrí medzi $\mathcal{NP}-complete$, takže nie je možné nájsť riešenie v polynomiálnom čase (ak platí $\mathcal{P} \neq \mathcal{NP}$). Pre riešenie sa používajú algoritmy ako *Simulated annealing*, alebo rôzne genetické algoritmy.

3 Matematický model

Cieľom tejto práce bolo vytvoriť matematický model, ktorý popisuje zvoz odpadu na úrovni obcí rôznymi prostriedkami a minimalizuje celkové náklady na prepravu. Výstupom z modelu je návrh miest výstavby prekladacích staníc a prechodu na železničnú dopravu. Možné modifikácie modelu sú uvedené na konci tejto kapitoly.

Organizácia železničnej dopravy je zložitejšia ako pri cestnej doprave. Pre porovnanie, v prípade cestnej nákladnej dopravy je potrebné zaplatiť mýto a nákladné auto môže vyraziť na cestu takmer v ľubovolnom čase. V prípade železnice je nutné vytipovať dopravnú cestu na základe parametrov železničnej súpravy, zaistiť si zakúpenie dopravnej cesty, teda možnosť prejazdu, cez prevádzkovateľa dráhy. Existujú dva typy žiadostí. Žiadosť o pridelenie kapacity do ročného jazdného plánu alebo žiadosť o pridelenie kapacity tzv. Ad hoc. Poplatok za dostupnosť dopravnej cesty v režime Ad hoc závisí na dobe vybavenia, tj. s akým časovým predstihom je žiadosť podaná. Pridelená dopravná cesta môže byť odobraná v prípade nevyužitia prejazdu a je nutné túto skutočnosť nahlásiť prevádzkovateľovi. Nevyužitie dopravnej cesty je spoplatnené. Taktiež je v prípade železničnej dopravy nutné zaplatiť poplatok za využitie dráhy na základe kategórie trate, zaťaženia a dalších parametrov. Detailný popis možno nájsť v Prohlášení o dráze, ktoré je vydávané organizáciu SŽDC (Správa železniční dopravní cesty) [7].

Vzhľadom k rýchlosti sa môže zdať železničná doprava výhodná, ale ak berieme do úvahy všetky nutné úkony (kontrola bŕzd, preloženie, vypravenie a ďalšie), priemerná rýchlosť sa pohybuje okolo 20km/h. Na úrovni Českej republiky je uprednostnená cestná nákladná doprava pred železničnou z dôvodu komplikácií akými sú realizácia prekládky taktiež infraštuktúra na území ČR v kontexte odpadového hospodárstva nie je zatiaľ rozvinutá. Ako výhody železničnej dopravy je nutné zmieniť, že na území Českej republiky nie je tento typ dopravy, až na výnimočné prípady, ovplyvnený poveternostnými podmienkami alebo veľkými sklonovými pomermi. V prípade využívania elektrických lokomotív vzniká minimum emisií a ich účinnosť je vyššia v porovnaní s dieselovými. Možno konštatovať, že železničná doprava je ekologickejšia ako doprava cestná. Ako typický príklad možno uviesť tranzitné miesta v Alpách, v ktorých je zakázaný prejazd ťažkým kamiónom a doprava je realizovaná pomocou železničnej dopravy. Kamióny sú naložené na železničné súpravy a prepravené cez hory. Takýchto prípadov je možné nájsť v Európe veľké množstvo.

3.1 Popis úlohy

Nech existuje množina uzlov, ktoré môžme rozdeliť na dva typy, producenti a spracovatelia. Množina producentov obsahuje podmnožinu miest, z ktorých sa uvažuje prevoz odpadu vlakom priamo k spracovaťeľovi. Úlohou je premiestniť všetok odpad produkovaný producentami k spracovateľom za najnižšiu cenu. Cena pozostáva z dopravy a ceny za spracovanie, ktorá môže byť pri rôznych spracovateľoch odlišná. Spôsobov dopravy je niekoľko. Najmenej ekonomickým a ekologickým spôsobom je prevoz pomocou, klasických automobilov pre zvoz SKO (směsný komunální odpad), ďalej označených ako KUKA. Alternatívou je použitie zvozovej súpravy. Odpad je však potrebné naložiť, prípadne vyložiť len v dopredu daných uzloch, v ktorých sa nachádza prekladacia stanica. Ďalšou možnosťou je využiť železničnú nákladnú dopravu, pre ktorú však znovu platí, že ju je možné využiť len vo vybraných uzloch. Poslednou možnosťou je vybudovať v niektorom uzle prekladaciu stanicu na železnicu s tým, že medzi týmto uzlom a spracovateľom bude odpad prevážaný pomocou nákladného vlaku, ďalej označeného ako kyvadlo. Tento vlak je určený výhradne na zvoz odpadu medzi uzlom a spracovateľom. Zjednodušenie v modeli je značné tým, že všetky ceny za dopravu sú uvažované lineárne vzhľadom k prejdeným kilometrom a množstvu prevezeného odpadu. Ďalej cenu za zriadenie uvažujeme konštantnú. V rámci preprocesingu je potrebné vypočítať nejkratšie vzdialenosti medzi mestami a spracovateľmi, ktoré vedú po železnici. Najkratšie cesty môžu byť spočítané napríklad pomocou Floyd-Warshall algoritmu alebo Jonsonovho algoritmu, ktorý je pre riedke grafy rýchlejší.

3.2 Označenie

V nasledujúcej časti bude zavedené značenie a vysvetlný význam parametrov, množín a premenných použitých v modeli.

Množiny

$\mathcal{M}$	množina všetkých uzlov,
$\mathcal{I}$	množina producentov,
$\mathcal{J}$	množina miest s uvažovaným zriadením kyvadla,
$\mathcal{K}$	množina spracovateľov,
$\mathcal{L}$	množina producentov, ktorý nepatria do možiny $\mathcal{J}$,
$\mathcal{Q}$	množina železničných hrán,
$\mathcal{R}$	množina cestných hrán,

Parametre

p_i	produkcia odpadu v uzle i [t],
C_k	cena za spracovanie tony odpadu v zariadení k [Kč/t],
c^K	náklady na prepravu tony odpadu na 1km pomocou KUKA [Kč/km]
c^Z	náklady na prepravu tony odpadu na 1km pomocou železničnej nákladnej dopravy [Kč/km]
c^A	náklady na prepravu tony odpadu na 1km pomocou zvozovej súpravy [Kč/km]
$c^O_{j,k}$	náklady na prepravu vagónu odpadu medzi j a k pomocou kyvadla [Kč]
v^K_r	dĺžka cestnej hrany r [km]
v^Z_q	dĺžka železničnej hrany q [km]
M	veľká konštanta zabezpečujúca správne fungovanie obmedzení [-]
t	kapaciata vagónu v tonách [t]
n	náklady na výstavbu železničnej prekladacej stanice [Kč]
b	maximálny počet vagónov prepraviteľných kyvadlom [-]

$IM^C_{i,r}$ $\begin{cases} 1 & \text{ak hrana } r \text{ vedie do uzla } i \\ -1 & \text{ak hrana } r \text{ vedie z uzla } i, \end{cases}$

$IM^Z_{i,q}$ $\begin{cases} 1 & \text{ak hrana } q \text{ vedie do uzla } i \\ -1 & \text{ak hrana } q \text{ vedie z uzla } i, \end{cases}$

z^Z_j $\begin{cases} 1 & \text{ak je v uzle } j \text{ možné prejsť na železničnú dopravu} \\ 0 & \text{inak,} \end{cases}$

z^C_i $\begin{cases} 1 & \text{ak je v uzle } i \text{ možné prejsť na nákladné autá} \\ 0 & \text{inak,} \end{cases}$

Premenné

s_k množstvo odpadu spracované v zariadení k [t],

$\delta_{j,k}$ $\begin{cases} 1 & \text{uvažuje sa zriadenie kyvadla medzi uzlami } j \text{ a } k \ (j \in \mathcal{J},\ k \in \mathcal{K}) \\ 0 & \text{inak,} \end{cases}$

$x_{j,k}$ počet vagónov prepravených kyvadlom z j do k [-]

a_r množstvo odpadu vezené po hrane r pomocou KUKA [t]

b_q množstvo odpadu vezené po hrane q pomocou železnice [t]

d_r množstvo odpadu vezené po hrane r pomocou zvozovej súpravy [t]

3.3 Model

$$min \sum_{k\in\mathcal{K}} C_k s_k + \sum_{r\in\mathcal{R}} (c^K a_r + c^A d_r) + \sum_{q\in\mathcal{Q}} c^Z b_q + \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}} (n\delta_{j,k} + x_{j,k} c^O_{j,k}) \tag{3.1}$$

$$-p_i = \sum_{r\in\mathcal{R}} IM^C_{i,r}(a_r + c_r) + \sum_{q\in\mathcal{Q}} IM^Z_{i,q} b_q + \sum_{k\in\mathcal{K}} x_{i,k} t \qquad \forall i \in \mathcal{I} \tag{3.2}$$

$$\sum_{k\in\mathcal{K}} \delta_{j,k} \leq 1 \qquad \forall j \in \mathcal{J} \tag{3.3}$$

$$x_{i,k} \leq \delta_{j,k} M \qquad \forall i \in \mathcal{I}, \forall k \in \mathcal{K} \tag{3.4}$$

$$-M z^Z_j - M \sum_{k\in\mathcal{K}} \delta_{j,k} \leq \sum_{q\in\mathcal{Q}} IM^Z_{j,q} b_q \qquad \forall j \in \mathcal{J} \tag{3.5}$$

$$M z^Z_j + M \sum_{k\in\mathcal{K}} \delta_{j,k} \geq \sum_{q\in\mathcal{Q}} IM^Z_{j,q} b_q \qquad \forall j \in \mathcal{J} \tag{3.6}$$

$$-M z^C_i \leq \sum_{q\in\mathcal{R}} IM^C_{i,r} d_r \qquad \forall i \in \mathcal{I} \tag{3.7}$$

$$M z^C_i \geq \sum_{q\in\mathcal{R}} IM^C_{i,r} d_r \qquad \forall i \in \mathcal{I} \tag{3.8}$$

$$\sum_{k\in\mathcal{K}} x_{j,k} \leq b \qquad \forall j \in \mathcal{J} \tag{3.9}$$

$$\sum_{k\in\mathcal{K}} x_{j,k} = 0 \qquad \forall j \in \mathcal{L} \tag{3.10}$$

$$s_k \geq 0 \qquad \forall k \in \mathcal{K} \tag{3.11}$$

$$\delta_{j,k} \in \{0,1\} \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K} \tag{3.12}$$

$$x_{j,k} \in \mathbb{N}_0 \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K} \tag{3.13}$$

$$a_r \geq 0 \qquad \forall r \in \mathcal{R} \tag{3.14}$$

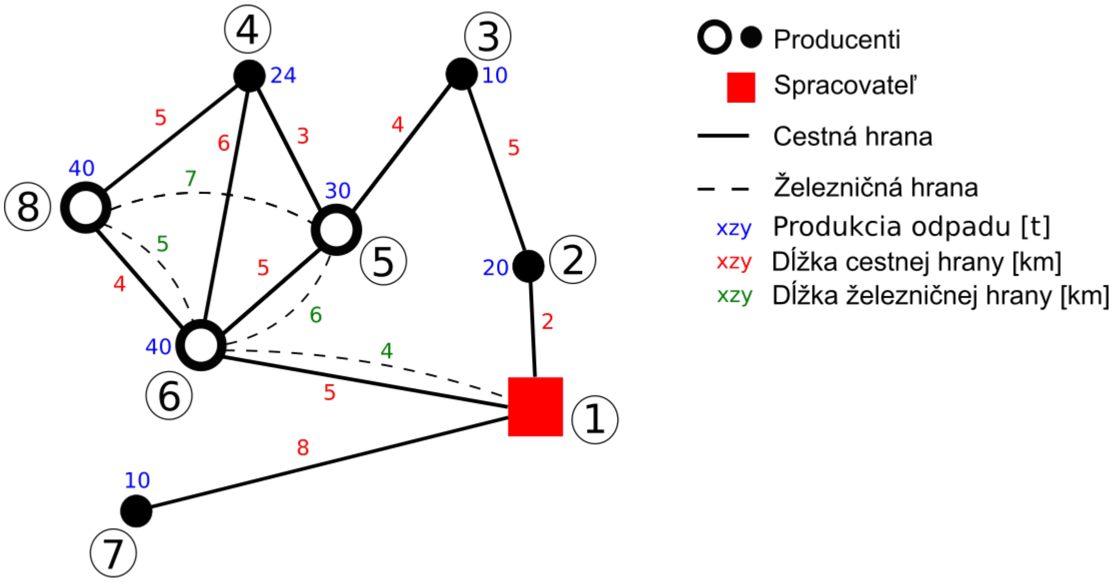
$$d_r \geq 0 \qquad \forall r \in \mathcal{R} \tag{3.15}$$

$$b_q \geq 0 \qquad \forall q \in \mathcal{Q} \tag{3.16}$$

Výstupom z modelu je množstvo odpadu prepravené po jednotlivých hranách a miesta, z ktorých je zriadené kyvadlo. Účelová funkcia (3.1) minimalizije celkové náklady na spracovanie a prevoz odpadu. Prvý člen funkcie predstavuje náklady na spracovanie všetkého vyprodukovaného odpadu. Druhý člen tvoria náklady na dopravu po ceste, tj. náklady na KUKA a zvozovú súpravu. Tretí člen predstavuje náklady na transport pomocou železnice, nie však kyvadlom. Posledný člen tvoria náklady na výstavbu prekladacej stanice a náklady spojené s transportom pomocou kyvadla. Rovnica (3.2) je bilančná rovnica uzlu. Rovnica (3.3) zabezpečuje, aby nebolo priradené jedno mesto so zriadeným kyvadlom k viacerým spracovateľom. Rovnica (3.4) umožňuje prevoz odpadu pomocou kyvadla. Rovnice (3.5) a (3.6) zamedzujú prechod na železničnú nákladnú dopravu v uzloch, kde to nie je možné. Rovnice (3.7) a (3.8) zamedzujú prechod na zvozovú súpravu v uzloch, kde to nie je možné. Rovnica (3.9) obmedzuje počet vagónov prepravených kyvadlom. Rovnica (3.10) zamedzuje prevoz kyvadlom z uzlov, kde nie je uvažovaná výstavba prekladacej stanice. Výpočetnú náročnosť MIP modelov pomocou exaktných metód ovplyvňujú predovšetkým celočíselné premenné. Výsledný model neobsahuje veľký počet celočíselných premenných ani pri rozsiahlejších dátach takže je prepoklad možného výpočtu exaktnými metódami v prijateľnom čase.

3.4 Aplikácia na malej úlohe

V tejto úlohe je uvažovaných sedem producentov odpadu a jeden spracovateľ. Z dôvodu veľkého množstva vstupných údajov budú uvedené v prílohe A. Celá situácia je zobrazená na Obr. 3.1.



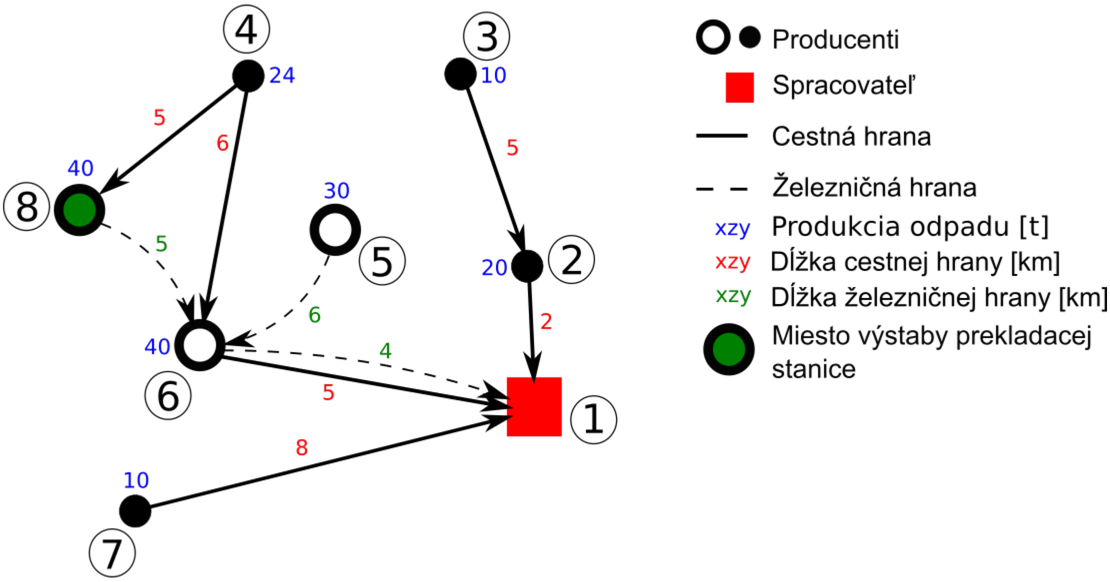
Obr. 3.1: Grafické zobrazenie malej úlohy

Uzol 1 predstavuje spracovateľa, ostatné uzly sú producentami. Uzly 5, 6, 8 sú ORP, patria do množiny $\mathcal{J}$. Cestné hrany sú zobrazené plnými čiarami a železničné hrany prerušovanými čiarami. Model bol implementovaný v GAMSe a výsledky je možné nájsť v tabuľke 3.1.

Tabuľka 3.1: Výsledné riešenie modelu

Množstvostvo odpadu [t]				kyvadlo		
Hrana	KUKA	Zvozová súprava	Železničná doprava	producent	zariadenie	počet vagónov
2-1	30	0	0	8	1	10
4-6	14	0	0			
7-1	10	0	0			
3-2	10	0	0			
4-8	10	0	0			
6-1	0	54	30			
5-6	0	0	30			

Po hrane 2-1, teda z uzlu 2 do uzlu 1 je vezených 30 ton odpadu. Kyvadlo je zriadené medzi producentom 8 a spracovateľom 1. Pomocou kyvadla je prevezených 10 vagónov odpadu z uzlu 8 k spracovateľovi 1. Na Obr. 3.2. sú zobrazené výsledné smery toku odpadu. V uzle číslo 6 je využitá možnosť preloženia z KUKA na zvozovú súpravu. Všetok odpad produkovaný v uzle 6 je vezený pomocou železničnej dopravy až k spracovateľovi. Hodnota účelovej funkcie je skreslená z dvôvodu malých vzdialeností v úlohe a prispôbenej ceny za zriadenie kyvadla a spracovanie odpadu. Jej hodnota je 7012 Kč. Náklady spojené s kyvadlom sú 2050 Kč z toho 700 Kč sú náklady na zriadenie.



Obr. 3.2: Grafické zobrazenie malej úlohy

3.5 Modifikácie modelu

Vytvorený model nepopisuje reálnu situáciu s dostatočnou presnosťou. Model možno priblížiť k realite ak budeme brať do úvahy nelineárnu závislosť ceny za prepravu odpadu po železnici na vzdialenosti a hmotnosti, $c^K(vzdialenos\check{t}, hmotnos\check{t})$. Ak sa pri modelovaní ceny budeme opierať o [3], tak cena predstavuje plochu v 3-rozmernom priestore. Keďže chceme v modeli zachovať linearitu, je potrebné nelineárnu funkciu c^K aproximovať po častiach lineárnou funkciou. Ďalším priblížením môže byť zavedenie ceny za prekládku odpadu z jedného spôsobu prepravy na druhý. Dá sa očakávať, že pre reálne situácie použitie železničnej dopravy nebude ekonomicky najvýhodnejším riešením, to však nemusí byť cieľom. Model môže obsahovať dalšie obmedzenia ktoré vynútia čiastočné využitie železničnej dopravy napríklad z dôvodu odľahčenia cestnej dopravy.

Záver

Cieľom práce bolo zoznámiť sa s problematikou dopravných úloh so zameraním na kombinatorické problémy a vytvoriť matematický model zvozu odpadu pomocou kombinovanej dopravy. Vytvorený model bol implementovaný v GAMSe.

Prvá časť práce pozostáva z úvodu do matematického programovania. Táto časť obsahuje všeobecné formulácie, zavedenie pojmov a možné spôsoby zápisu. Na vzorový príklad boli uplatnené dve metódy riešenia, grafické riešenie a simplexová metóda. Druhá časť je venovaná vybraným modelom celočíselného programovania. Každý model je popísaný, je uvedená jedna z možných formulácií a algoritmy používané pre výpočet. Posledná časť obsahuje stručne výhody a nevýhody spojené so zeležničnou nákladnou dopravou a popis úlohy pre ktorú bol následne formulovaný matematický model.

Model zostavený na základe poznatkov z predchádzajúcich kapitol bol testovaný na malej úlohe. Výsledky sú návrhom pre organizáciu zvozu odpadu a výstavbu prekladacích staníc, ktoré umožnia lepšie zapojenie železničnej dopravy do procesu. Je možné ich nájsť v tabuľke 3.1 a na Obr. 3.2. Na záver sú uvedené nedostatky modelu, možnosti úpravy a rozšírenia, ktoré by mohli priblížiť model skutočnosti.

Literatúra

[1] CABALKA, M. *Celočíselná optimalizace pro řešení dopravních úloh.* Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2016. 46 s. Vedoucí RNDr. Pavel Popela, Ph.D.

[2] DEMEL, Jiří. *Grafy a jejich aplikace.* Praha: Academia, 2002. ISBN 80-200-0990-6.

[3] GAL, P. *Doprava komunálních odpadů.* Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2013. 55 s. Vedoucí Ing. Martin Pavlas, Ph.D.

[4] KU, W. BECK, J. C. Mixed Integer Programming models for job shop scheduling: A computational analysis. *Computers & Operations Research*, 73, 2016, p.165-173. ISSN 0305-0548.

[5] NEVRLÝ, V. *Modely a metody pro svozové úlohy.* Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2016. 48 s. Vedoucí RNDr. Pavel Popela Ph.D.

[6] PATAKI, G. Teaching Integer programming Formulations using the Travelling Salesman Problem. *SIAM Review*, 45, 2003, p.116-123. ISSN 00361445.

[7] SŽDC. *Prohlášenío dráze celostátní a regionální* SŽDC [online]. 2018 [citováno 2018-05-23]. Dostupné z: http://www.szdc.cz/soubory/ /prohlaseni-o-draze/2019/05-12-17-szdc-prohlaseni-o-draze-2019.pdf.

[8] WILLIAMS, H. P. *Model Building in Mathematical Programming.* 5th ed. Chichester: John Wiley, 1978. ISBN 978-1-118-44333-0.

[9] WOLSEY, L. A. *Integer Programming.* 1st ed. New York: John Wiley, 1998. ISBN 978-0471283669.

Zoznam použitých skratiek a symbolov

IP	Integer Programming
MIP	Mixed Integer Programming
GAMS	General Algebraic Modeling System
BP	Binary Programming
PIP	Pure Integer Programming
LP	Linear Programming
JSP	Job-Shop Problem
MTZ	Miller-Tucker-Zemlin
SŽDC	Správa železniční dopravní cesty
SKO	směsný komunální odpad

A Kód matematického modelu v GAMSe a vstupné dáta pre model

Kód matematického modelu a vstupné dáta sú uvedené na CD.